CUDA Implementation of Position Based Fluids

CSC417 Course Project

Qingyuan Qie, Changlin Su

Department of Computer Science
University of Toronto

December 20, 2021

Enforcing incompressibility

For particle i at position p_i, we compute the density of the fluid around particle i using the estimator:
$\rho_{F(i)} = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Enforcing incompressibility

For particle i at position $\boldsymbol{p}_i$, we compute the density of the fluid around particle i using the estimator:
$\rho_i = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Then we have the constant density constraint:
$C_i(\boldsymbol{p}) = \frac{\rho_i}{\rho_0} - 1$

Enforcing incompressibility

For particle i at position $\boldsymbol{p}_i$, we compute the density of the fluid around particle i using the estimator:
$\rho_i = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Then we have the constant density constraint:
$C_i(\boldsymbol{p}) = \frac{\rho_i}{\rho_0} - 1$

And we want to compute a position correction $\Delta \boldsymbol{p}$, such that:
$C(\boldsymbol{p} + \Delta \boldsymbol{p}) = 0$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,

$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,
$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Then the position correction $\Delta \boldsymbol{p}_i$ including affect from neighboring particles is,
$\Delta \boldsymbol{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,
$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Then the position correction $\Delta \boldsymbol{p}_i$ including affect from neighboring particles is,
$\Delta \boldsymbol{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

And the position update is,
$\boldsymbol{p}_i^* = \boldsymbol{p}_i + \Delta \boldsymbol{p}_i$

Simulation Step

Algorithm 1 simulation step

```
1:  for all particles i do:                                   ▷ Fluid advect
2:      apply forces $\boldsymbol{v}_i = \boldsymbol{v}_i + \Delta t f_{ext}$
3:      predict position $\boldsymbol{x}_i^* = \boldsymbol{x}_i + \Delta t \boldsymbol{v}_{\boldsymbol{i}}$
4:  end for
5:  for all particles i do:
6:      find neighboring particles $F(\boldsymbol{x}_i^*)$
7:  end for
8:  while $iter < solverIterations$ do:                       ▷ Iterativly solves
    imcompressibility constraints
9:      for all particles i do:
10:         calculate $\lambda_i$
11:     end for
12:     for all particles i do:
13:         calculate $\Delta \boldsymbol{p}_i$
14:         perform collision detection and response
15:     end for
16:     for all particles i do:
17:         update position $\boldsymbol{x}_i^* = \boldsymbol{x}_i^* + \Delta \boldsymbol{p}_i$
18:     end for
19: end while
20: for all particles i do:
21:     update velocity $\boldsymbol{v}_i = \frac{1}{\Delta t}(\boldsymbol{x}_i^* - \boldsymbol{x}_i)$
22:     apply viscosity
23:     update position $\boldsymbol{x}_i = \boldsymbol{x}_i^*$
24: end for
```

Performance of Implementation

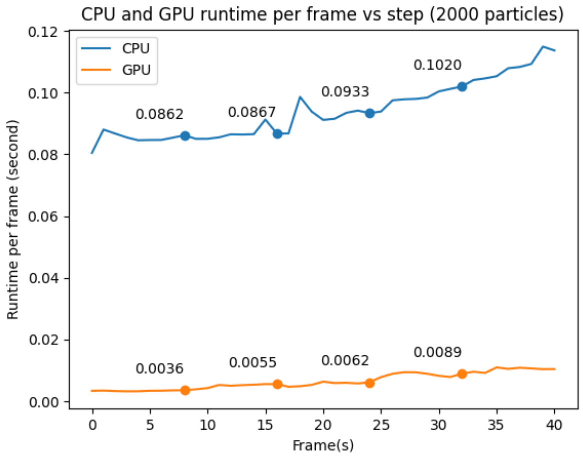

Figure: Performance Comparison between CPU ans GPU, 2k particles

Performance of CUDA Implementation

Particle Count	Time per Frame	Frame per Second(fps)
2k	0.006s	166fps
12k	0.01s	100fps
27k	0.021s	47.6fps
80k	0.064	15.6fps